Extraction of Text Lines and Text Blocks on Document Images Based on Statistical Modeling

Su Chen* **and Robert M. Haralick**

Department of Electrical Engineering, University of Washington, Seattle, WA 98195

Ihsin T. Phillips

Department of Computer Science, Seattle University, Seattle, WA 98122

ABSTRACT

In this article, we developed a Bayesian model to characterize text line and text block structures on document images using the text word bounding boxes. We posed the extraction problem as finding the text lines and text blocks that maximize the Bayesian probability of the text lines and text blocks given the text word bounding boxes. In particular, we derived the so-called probabilistic linear displacement model (PLDM) to model the text line structures from text word bounding boxes. We also developed an augmented PLDM model to characterize the text block structures from text line bounding boxes. By systematically gathering statistics from a large population of document images, we are able to validate our models through experiments and determine the proper model parameters. We designed and implemented an iterative algorithm that used these probabilistic models to extract the text lines and text blocks. The quantitative performances of the algorithm in terms of the rates of miss, false, correct, splitting, merging, and spurious detections of the text lines and text blocks are reported. © 1996 John Wiley & Sons, Inc.

I. INTRODUCTION

Document layout analysis identifies various objects of interest on a document image and describes their spatial relations. An object is defined as a homogeneous rectangular region that corresponds to one type: character, word, text line, text block, or nontextual region.

Earlier work on document layout analysis can be categorically divided into two groups. One group employs the top-down or model-driven approach [1,2,3]. It starts at the global image level and successively decomposes the image into smaller regions. Each region has one type: character, word, text line, text block, or nontextual region. Nagy [2] and Srihari [3] employed an X–Y tree as the representation of a document layout structure. The X–Y tree is a nested decomposition of rectangular blocks into smaller rectangular blocks. Each node in the X–Y tree corresponds to a rectangular block. The root node is the largest rectangular block, i.e., the input document image. At each level, the decomposition is induced by partitions only in one direction (horizontal or vertical), but a block may have an arbitrary number of children. In the process of partitioning, a block is segmented into subblocks by making cuts in the horizontal profile corresponding to troughs of depth and width greater than some threshold. Each resulting subblock has a vertical projection profile that can be similarly partitioned for vertical segmentation. The segmentation process may be carried out recursively to any desired depth with alternating horizontal and vertical subdivisions.

The main problems of this approach are as follows. 1) At each step of the successive decompositions, the system has to select the correct decomposition model since the models for the text block, text line, word, or character, decomposition are inherently different. On the other hand, there are occasions when such model selections do not correspond to the levels in the decomposition tree. 2) Some popular top-down decomposition schemes, such as the above-mentioned recursive X–Y cut technique, do not work for certain types of document layout topology.

The other group adopts the bottom-up or data-driven approach [4,5]. It starts by synthesizing evidence at the black-and-white pixel level and then merges pixels into characters, characters into words, words into lines, lines into blocks, etc., until the whole document is completely labeled [4]. The technique is based on a connected component analysis. A connected component is a set of binary one pixels in a binary image which are either four-connected or eight-connected. The algorithm assumes that each connected component in the image corresponds to one character or nontextual object. It starts by extracting all the connected components in the input image. A Hough transform is applied to the centroid of the enclosing rectangles of the connected components to find collinear components. Positional relationships between collinear components, an intercharacter gap threshold, and an interword gap threshold are then used to group the components into text strings. One drawback of the method is that it is sensitive to touching characters and fragmented characters because the underlying connectivity assumptions are violated.

The problems associated with all these earlier techniques are that they were developed on a trial-and-error basis and although they provide illustrative results, hardly any have been tested on significant sized data sets. In addition, most papers do not give any explicit quantitative performance measure of their system. Al-

Revised manuscript received 29 May 1996

* Present address: Caere Corporation, 100 Cooper Court, Los Gatos, CA 95030, USA.

International Journal of Imaging Systems and Technology, Vol. 7, 343–356 (1996)
© 1996 John Wiley & Sons, Inc. CCC 0899–9457/96/040343–14

though the appropriate performance measures for the document layout analysis are not obvious and are hard to derive, it is clear that suitable performance measures not only facilitate us to construct a system that optimizes its performance measures given the training data set, but also enable us to predict the system performance based on the testing data set.

This article describes the continuation of work reported in [10] and [11], where we developed and evaluated a word segmentation algorithm that is capable of detecting all text words on document images simultaneously. We derived quantitative measures to evaluate the performance of the document layout analysis algorithms. The experimental results indicated that we achieved a high correct word detection rate (about 95%) over a very large image population. In this work, we develop statistical models to characterize the text line and text block structures on document images given the text word bounding boxes. We pose the extraction problem as finding the text lines and text blocks that maximize the Bayesian probability of the text lines and text blocks by observing the text word bounding boxes.

Section II provides the general problem statement of the text line and text block detection and describes an iterative algorithm to solve the problem. Section III discusses the probabilistic linear displacement model (PLDM) and demonstrates how it can be used to model the text line structures on document images. Section IV describes a generic Bayesian algorithm for detecting the linear displacement structures (LDS) from a set of observations. Then in Section V, we apply the generic algorithm to extract text lines from the document images. Section VI describes an augmented probabilistic linear displacement model (APLDM) for characterizing the text block structures on document images. Section VII uses the APLDM model to extract text block from the document images. Finally, in Section VIII, we discuss an experimental protocol to validate our models via experiments, and reports the performance of our algorithms.

II. TEXT LINE AND TEXT BLOCK DETECTION

Let $\hat{\Sigma}$ denote the set of detected word bounding boxes from a bilevel document image I. Let Φ and Λ be the sets of text lines and text blocks on I. The problem of text line and text block extraction can be formulated as follows: Given a set of detected word bounding boxes $\hat{\Sigma}$, find the Φ and Λ that maximize the conditional probability $P(\Phi, \Lambda \mid \hat{\Sigma})$.

Based on the above Bayesian formulation, the conditional probability $P(\Phi, \Lambda \mid \hat{\Sigma})$ can be evaluated as:

$$P(\Phi, \Lambda \mid \hat{\Sigma}) = P(\Lambda \mid \hat{\Sigma}, \Phi) P(\Phi \mid \hat{\Sigma}) \tag{1}$$

$$= P(\Phi \mid \hat{\Sigma}, \Lambda) P(\Lambda \mid \hat{\Sigma}) . \tag{2}$$

Suppose we have an initial estimate of (Φ, Λ), denoted by (Φ^t, Λ^t), where $t = 0$. Normally, we would start Φ^0 as the single bounding box that encloses every word bounding box in $\hat{\Sigma}$ and $\Lambda^0 = \emptyset$. Then we can iteratively update the Φ^t and Λ^t to maximize the conditional probability $P(\Phi, \Lambda \mid \hat{\Sigma})$ via the following two-step process (let N_{iter} be the number of iterations):

Algorithm 1. Text Line and Text Block Detection

1. Let $t = 0$.
2. Computes the optimal Λ^{t+1} such that:

$$\Lambda^{t+1} = arg \max_{\Lambda} P(\Phi = \Phi^t, \Lambda \mid \hat{\Sigma})$$

$$= arg \max_{\Lambda} P(\Lambda \mid \hat{\Sigma}, \Phi^t) . \tag{3}$$

We will describe an algorithm to compute Λ^{t+1} in Section V.

3. Calculates the optimal Φ^{t+1} such that:

$$\Phi^{t+1} = arg \max_{\Phi} P(\Phi, \Lambda = \Lambda^{t+1} \mid \hat{\Sigma})$$

$$= arg \max_{\Phi} P(\Phi \mid \Lambda^{t+1})$$

$$= arg \max_{\Phi} P(\Phi \mid \Lambda^{t+1}) \tag{4}$$

where it is assumed that $P(\Phi \mid \hat{\Sigma}, \Lambda^{t+1}) = P(\Phi \mid \Lambda^{t+1})$. We will describe an algorithm to compute Φ^{t+1} in Section VII.

4. If $t \leq N_{iter}$ then return (Φ^t, Λ^t); else $t = t + 1$ and go to Step 2. □

In the following sections, we begin to introduce the statistical models to characterize the text line and text block structures on document images (see Sections III and VI). Then, in Section V, we describe the solution to the first subproblem of finding the text lines given an initial delineation of text blocks and a set of text word bounding boxes. In Section VII, we solve the second subproblem of finding the text blocks given a set of text line bounding boxes.

III. PROBABILISTIC LINEAR DISPLACEMENT MODEL

We first describe the deterministic linear displacement model (DLDM), and then generalize it to the probabilistic case. In the DLDM, we have a set of M objects ($M \geq 2$), denoted by $\mathcal{B} = \{B_1, B_2, \ldots, B_M\}$. Let L denote a baseline of these objects, which is represented by the equation:

$$x \sin \varphi + y \cos \varphi - \rho = 0 \tag{5}$$

where φ denotes the orientation of the baseline and ρ is the distance of the baseline to the origin of the coordinate system.

The DLDM requires that the objects $B_1, B_2, \ldots, B_M$ are collinear along the baseline L. Let $\epsilon_i = \epsilon(B_i, L)$ define a distance function of B_i to the baseline L and $i = 1, 2, \ldots, M$. Then, the collinearity implies $\epsilon_i = 0$, where $i = 1, 2, \ldots, M$.

Since $B_1, B_2, \ldots, B_M$ are collinear, we can order them according to their positions along the baseline L. Without the loss of generality, we would assume that the sequence $(B_1, B_2, \ldots, B_M)$ is one such an ordering, denoted as $B_1 \leq B_2 \leq \cdots \leq B_M$. The DLDM requires that the adjacent objects in the sequence $(B_1, B_2, \ldots, B_M)$ be equally spaced along the baseline L. Let $\delta_i = \delta(B_i, B_{i+1})$ define a distance function between B_i and B_{i+1} along the baseline L (also defined as the displacement), where $i = 1, 2, \ldots, M - 1$. Figure 1 shows an example of the DLDM.

In general, if a set of M objects can be modeled by a linear

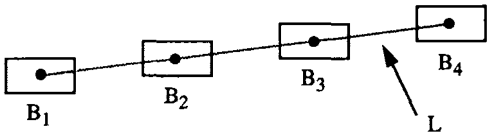


Figure 1. Example of the DLDM. The geometric centroids of B_1, B_2, B_3, and B_4 are collinear and equally spaced along the baseline L.

displacement model, we say that the set of objects constitute a *linear displacement structure* (LDS). We also call the pair (B_i, B_{i+1}) adjacent in the LDS, where $i = 1, 2, \ldots, M-1$. When $M = 1$, we treat the object as a special case of LDS with a single object.

In contrary to the DLDM, which requires $B_1, B_2, \ldots, B_M$ to be strictly collinear along the baseline L (i.e., $\epsilon_i = 0$ for $i = 1, 2, \ldots, M$) and the displacement to be a constant (i.e., $\delta_i = \mu_\delta$ for $i = 1, 2, \ldots, M-1$), the PLDM says that they follow a probability distribution. Typically, we would assume that ϵ_i $(i = 1, 2, \ldots, M)$ and δ_i $(i = 1, 2, \ldots, M-1)$ are independently distributed random variables given the baseline (ρ, φ) and the displacement μ_δ, as indicated through the following probability calculations:

$$\begin{aligned}
&P(B_1, B_2, \ldots, B_M \mid \rho, \varphi, \mu_\delta) \\
&\quad = P(\epsilon_1, \epsilon_2, \ldots, \epsilon_M, \delta_1, \delta_2, \ldots, \delta_{M-1} \mid \rho, \varphi, \mu_\delta) \\
&\quad = P(\epsilon_1, \epsilon_2, \ldots, \epsilon_M \mid \rho, \varphi, \mu_\delta) P(\delta_1, \delta_2, \ldots, \delta_{M-1} \mid \rho, \varphi, \mu_\delta) \\
&\quad = P(\epsilon_1, \epsilon_2, \ldots, \epsilon_M \mid \rho, \varphi) P(\delta_1, \delta_2, \ldots, \delta_{M-1} \mid \rho, \varphi, \mu_\delta) \\
&\quad = \prod_{i=1}^{M} P(\epsilon_i \mid \rho, \varphi) \prod_{i=1}^{M-1} P(\delta_i \mid \rho, \varphi, \mu_\delta) \,. \qquad (6)
\end{aligned}$$

We would further assume that $P(\epsilon_i \mid \rho, \varphi)$ and $P(\delta_i \mid \rho, \varphi, \mu_\delta)$ are Gaussian, i.e.,

$$\begin{aligned}
P(\epsilon_i \mid \rho, \varphi) &= \frac{1}{\sqrt{2\pi}\sigma_\epsilon} e^{-\frac{\epsilon_i^2}{2\sigma_\epsilon^2}} \,, \\
P(\delta_i \mid \rho, \varphi, \mu_\delta) &= \frac{1}{\sqrt{2\pi}\sigma_\delta} e^{-\frac{(\delta_i - \mu_\delta)^2}{2\sigma_\delta^2}}
\end{aligned} \qquad (7)$$

where μ_δ is an unknown mean and σ_ϵ, σ_δ are known constants. We provide an experimental study on the adequacy of these models in Section VIII.

In addition, we also could put a prior probability distribution on the parameters ρ, φ, and μ_δ. Using a model that the parameters are independent, we assume that ρ comes from a uniform distribution over an interval of size $1/\lambda$, i.e., $P(\rho) = \lambda$, and that φ, μ_δ come from Gaussian distributions

$$P(\varphi) = \frac{1}{\sqrt{2\pi}\sigma_\varphi} e^{-\frac{\varphi^2}{2\sigma_\varphi^2}} \,, \qquad P(\mu_\delta) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(\mu_\delta - \mu)^2}{2\sigma^2}} \qquad (8)$$

where σ_φ, μ, and σ are known constants.

It is now possible to write the joint probability distribution of $B_1, B_2, \ldots, B_M$ and ρ, φ, μ_δ given that $B_1, B_2, \ldots, B_M$ come from a linear displacement structure (LDS). Hence, we have

$$\begin{aligned}
&P(B_1, B_2, \ldots, B_M, \rho, \varphi, \mu_\delta \mid \text{from LDS}) \\
&\quad = \mathrm{P}(\mathrm{B}_1, \mathrm{B}_2, \ldots, \mathrm{B}_M \mid \rho, \varphi, \mu_\delta) \mathrm{P}(\rho, \varphi, \mu_\delta) \\
&\quad = \prod_{i=1}^{M} \mathrm{P}(\epsilon_i \mid \rho, \varphi) \prod_{i=1}^{M-1} \mathrm{P}(\delta_i \mid \rho, \varphi, \mu_\delta) \mathrm{P}(\rho) \mathrm{P}(\varphi) \mathrm{P}(\mu_\delta) \,. \qquad (9)
\end{aligned}$$

Figure 2 defines the six types of bounding box edges, where AB, EF, and CD are the top, center, and bottom horizontal edges; AD, GH, and BC are the left, center, and right vertical edges. The PLDM can be used to model separately the top, center, or bottom edges of the word-bounding boxes that belong to the same horizontal text line.

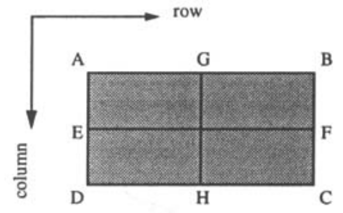


Figure 2. Six types of bounding box edges.

IV. GENERIC LINEAR DISPLACEMENT STRUCTURE DETECTION

In this section, we first pose the generic linear displacement structure detection problem. Then, we provide the theoretical analysis and describe a generic two-step algorithm for finding one linear displacement structure from a set of observed word edges. Finally, we describe an integrated algorithm for detecting multiple linear displacement structures from a set of observed word edges.

A. Generic Problem Statement. Let $\mathcal{E} = \{E_1, E_2, \ldots, E_M\}$ denote a set of observed word edges. Without the loss of generality, we assume that the edges are horizontal.

There exists a subset of $\mathcal{E}$, denoted by $\mathcal{S} = \{E_{i_1}, E_{i_2}, \ldots, E_{i_N}\} \subseteq \mathcal{E}$ that comes from a linear displacement structure L, where $N \leq M$ is the number edges in $\mathcal{S}$. Denote $\bar{\mathcal{S}} = \mathcal{E} - \mathcal{S}$. Let $\mathcal{J} = \{k \in \{1, 2, \ldots, M\} \mid E_k \in \mathcal{S}\}$ represent the set of edge indices in $\mathcal{S}$. Let the number of elements in $\mathcal{S}$ and $\bar{\mathcal{S}}$ be N and $M - N$, respectively. Without the loss of generality, we assume that $\mathcal{S} = \{E_{i_1}, E_{i_2}, \ldots, E_{i_N}\}$ is properly ordered along the baseline of L. Let $\delta_j = \delta(E_{i_j}, E_{i_{(j+1)}})$ for $j = 1, 2, \ldots, N-1$.

As in Section III, let the linear displacement structure L be parameterized by ρ, φ, and μ_δ. Then, the problem of detecting the linear displacement structure L by observing the edge set $\mathcal{E}$ can be formally stated as follows. Given the edge set $\mathcal{E} = \{E_1, E_2, \ldots, E_M\}$. Find the linear displacement structure L to maximize the conditional probability $P(\rho, \varphi, \mu_\delta, \mathcal{J} \mid \mathcal{E})$.

B. Algorithm Description. To set up the Bayesian framework, we need to write expressions for the probability density of observing $\mathcal{S}$ given that it comes from L. From Section III, we have

$$\begin{aligned}
P(\mathcal{S} \mid \text{from } L) &= P(\mathcal{S} \mid \rho, \varphi, \mu_\delta) \\
&= \prod_{i \in \mathcal{J}} P(\epsilon_i \mid \rho, \varphi) \prod_{j=1}^{N-1} P(\delta_j \mid \rho, \varphi, \mu_\delta) \,. \qquad (10)
\end{aligned}$$

Also, we need the probability density of observing $\bar{\mathcal{S}}$ given that it does not come from L. Using the model that the observations in $\bar{\mathcal{S}}$ are independent given that they do not come from L and that they come from a uniform distribution over an area of $1/\gamma$, we have

$$P(\bar{\mathcal{S}} \mid \text{not from } L) = \gamma^{M-N} \,. \qquad (11)$$

Finally, we need the probability q that an observed edge $E \in \mathcal{E}$ comes from the linear displacement structure L. Hence, $P(E \in \mathcal{S}) = q$ and $P(E \in \bar{\mathcal{S}}) = 1 - q$. By assuming $P(\mathcal{J} \mid \rho, \varphi, \mu_\delta) = P(\mathcal{J}) = q^N (1-q)^{M-N}$, we have

$$
\begin{aligned}
&P(\rho, \varphi, \mu_\delta, \mathscr{I} \mid \mathscr{E}) \\
&= \frac{P(\mathscr{E} \mid \mathscr{I}, \rho, \varphi, \mu_\delta)P(\mathscr{I} \mid \rho, \varphi, \mu_\delta)P(\rho)P(\varphi)P(\mu_\delta)}{P(\mathscr{E})} \\
&= \frac{P(\mathscr{S} \mid \rho, \varphi, \mu_\delta)P(\bar{\mathscr{S}} \mid \rho, \varphi, \mu_\delta)P(\mathscr{I})P(\rho)P(\varphi)P(\mu_\delta)}{P(\mathscr{E})} \\
&= \frac{P(\mathscr{S} \mid \rho, \varphi, \mu_\delta)P(\bar{\mathscr{S}})P(\mathscr{I})P(\rho)P(\varphi)P(\mu_\delta)}{P(\mathscr{E})} \\
&= \frac{\Pi_{i \in \mathscr{I}} P(\epsilon_i \mid \rho, \varphi) \Pi_{j=1}^{N-1} P(\delta_j \mid \rho, \varphi, \mu_\delta) q^N [\gamma(1-q)]^{M-N} P(\rho)P(\varphi)P(\mu_\delta)}{P(\mathscr{E})} .
\end{aligned}
\tag{12}
$$

By taking logarithms on both sides of the above equation, we can show that the maximization of the conditional probability $P(\rho, \varphi, \mu_\delta, I \mid \mathscr{E})$, is equivalent to finding the $\hat{\mathscr{I}}$, $\hat{\rho}$, $\hat{\varphi}$, and $\hat{\mu}_\delta$ to minimize the quantity J, where

$$
\begin{aligned}
J(\mathscr{I}, \rho, \varphi, \mu_\delta \mid \mathscr{E}) = \sum_{i \in \mathscr{I}} \frac{\epsilon_i^2}{2\sigma_\epsilon^2} + \sum_{i \in \mathscr{I}} \frac{(\delta_i - \mu_\delta)^2}{2\sigma_\delta^2} + \frac{\varphi^2}{2\sigma_\varphi^2} + \frac{(\mu_\delta - \mu)^2}{2\sigma^2} \\
- N \ln q - (M - N) \ln \gamma(1 - q) .
\end{aligned}
\tag{13}
$$

We adopt a two-step procedure to search for the minimum of J. The first step computes an optimal subset of collinear edges in $\mathscr{E}$, denoted by $(\mathscr{I}', \hat{\rho}, \hat{\varphi})$, that minimizes J_1, where

$$
\begin{aligned}
J_1(\mathscr{I}, \rho, \varphi \mid \mathscr{E}) = \sum_{i \in \mathscr{I}} \frac{\epsilon_i^2}{2\sigma_\varphi^2} + \frac{\varphi^2}{2\sigma_\varphi^2} - N \ln q \\
- (M - N) \ln \gamma(1 - q) .
\end{aligned}
\tag{14}
$$

Algorithm 2 summarizes the iterative minimization process (see Bayesian fitting [13]). It starts with an initial estimate of a baseline, denoted by (ρ^0, φ^0). We will describe a process to obtain this initial estimate in Section IVE.

Algorithm 2. Collinear Edge Detection

1. At iteration t, determine which collinear edges to take.

$$
\mathscr{I}^t = \{i \mid P(\epsilon_i \mid \rho^t, \varphi^t)q > \gamma(1-q), \text{ where } i = 1, 2, \ldots, M\} .
\tag{15}
$$

2. Compute the joint probability P^t. Define P^t by

$$
P^t = \prod_{i \in \mathscr{I}^t} P(\epsilon_i \mid \rho^t, \varphi^t) q^{N^t} [\gamma(1-q)]^{M-N^t} P(\rho^t)P(\varphi^t)
\tag{16}
$$

   where N^t is the size of $\mathscr{I}^t$.

3. Determine the new Bayesian estimate of the baseline parameters $(\rho^{t+1}, \varphi^{t+1})$ to maximize the probability (see Section IVC)

$$
P(\rho^{t+1})P(\varphi^{t+1}) \prod_{i \in \mathscr{I}^t} P(\epsilon_i \mid \rho^{t+1}, \varphi^{t+1}) .
\tag{17}
$$

4. Iterate as long as $P^{t+1} > P^t$ and $t < Q$, where Q is the maximum number of iterations.
5. Choose $\mathscr{I}' = \mathscr{I}^t$, $N' = N^t$, $\hat{\rho} = \rho^{t+1}$ and $\hat{\varphi} = \hat{\varphi}^{t+1}$. □

In the second step (see Algorithm 3), we search for the equally spaced edges in $\mathscr{S}' = \{E_i \in \mathscr{E} \mid i \in \mathscr{I}'\}$ given the optimal subset of collinear edges $(\mathscr{I}', \hat{\rho}, \hat{\varphi})$ from the previous step (assuming $N' \geq 2$), and calculates $\hat{\mathscr{I}}$ and $\hat{\mu}_\delta$ that minimize J_2, where

$$
\begin{aligned}
J_2(\mathscr{I}, \mu_\delta \mid \mathscr{E}, \mathscr{I}', \hat{\rho}, \hat{\varphi}) = \sum_{i \in \mathscr{I}} \frac{(\delta_i - \mu_\delta)^2}{2\sigma_\delta^2} + \frac{(\mu_\delta - \mu)^2}{2\sigma^2} \\
- N \ln q - (N' - N) \ln \gamma(1 - q) .
\end{aligned}
\tag{18}
$$

Algorithm 3. Equal Displacement Edge Detection

1. If $N' < 2$, then return $\hat{\mathscr{S}} = \emptyset$ and $\hat{\mathscr{I}} = \emptyset$.
2. Else order the edges in $\mathscr{S}'$ along the baseline $(\hat{\rho}, \hat{\varphi})$. Without the loss of generality, let $\mathscr{S}' = (E_{i_1}, E_{i_2}, \ldots, E_{i_{N'}})$ be the ordered sequence.
3. For each possible subsequence of edges of length $N + 1$, denoted by $\mathscr{S}(u, N) = (E_{i_u}, E_{i_{(u+1)}}, \ldots, E_{i_{(u+N)}})$, where $1 \leq u < u + N \leq N'$, derive the Bayesian estimate of the displacement $\mu_\delta = \hat{\mu}_\delta$ (see Section IVD) and compute the joint probability $P(u, N)$, defined by

$$
\begin{aligned}
P(u, N) = \prod_{j=1}^{N} P(\delta_j \mid \hat{\rho}, \hat{\varphi}, \hat{\mu}_\delta) q^N \\
\cdot [\gamma(1-q)]^{N'-N} \cdot P(\hat{\mu}_\delta)
\end{aligned}
\tag{19}
$$

   where $\delta_j = \delta(E_{i_{(u+j-1)}}, E_{i_{(u+j)}})$.
4. Determine the u^*, N^* that maximize $P(u, N)$.
5. Let $P_0 = [\gamma(1-q)]^{N^*}$ denote the probability that there is not a linear displacement structure in $\mathscr{S}(u^*, N^*)$ and $P^* = \Pi_{j=1}^{N^*} P(\delta_j \mid \hat{\rho}, \hat{\varphi}, \hat{\mu}_\delta) q^{N^*}$ denote the probability that $\mathscr{S}(u^*, N^*)$ comes from a linear displacement structure. If $P^* > P_0$, then return $\hat{\mathscr{S}} = \mathscr{S}(u^*, N^*)$, $\hat{\mathscr{I}} = \{k \mid E_k \in \hat{\mathscr{S}}\}$, and the corresponding $\hat{\mu}_\delta$; else return $\hat{\mathscr{S}} = \emptyset$ and $\hat{\mathscr{I}} = \emptyset$. □

Therefore, given a set of edges $\mathscr{E}$ and an initial baseline estimate (ρ^0, φ^0), the sequential execution of Algorithms 2 and 3 produces a single linear displacement structure in $\mathscr{E}$ (if there is one), parameterized by $(\hat{\rho}, \hat{\varphi}, \hat{\mu}_\delta, \hat{\mathscr{I}})$, that maximizes the conditional probability $P(\rho, \varphi, \mu_\delta, \mathscr{I} \mid \mathscr{E})$.

C. Baseline Parameter Estimation. In this section, we derive the Bayesian estimates of the baseline parameters (denoted by $\hat{\rho}$ and $\hat{\varphi}$) given a set of N collinear edges, denoted by $E_1, E_2, \ldots, E_N$. We want to compute the $\hat{\rho}$ and $\hat{\varphi}$ that maximize the joint probability distribution

$$
P(E_1, E_2, \ldots, E_N, \rho, \varphi) = P(\rho)P(\varphi) \prod_{i=1}^{N} P(\epsilon_i \mid \rho, \varphi) .
\tag{20}
$$

Using the PLDM assumptions (Section III), it is equivalent to minimizing the following sum of squares,

$$
Q(\rho, \varphi) = \sum_{i=1}^{N} \frac{\epsilon_i^2}{2\sigma_\epsilon^2} + \frac{\varphi^2}{2\sigma_\varphi^2} .
\tag{21}
$$

We need to first define the distance of E_i to the baseline L (denoted by ϵ_i, where $i = 1, 2, \ldots, N$). Figure 3 illustrates a horizontal edge E_i and a baseline, denoted by L. Let E_i be parameterized by its centroid (x_i, y_i) and length r_i. The ϵ_i is defined through the equation

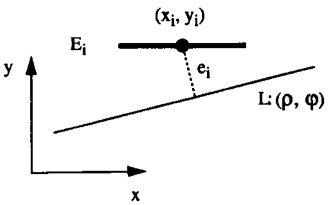


Figure 3. Edge E_i and its baseline L. The e_i is the distance of the centroid of E_i to the baseline.

$$\begin{aligned}\epsilon_i^2 &= \frac{1}{r_i}\int_{-r_i/2}^{r_i/2} [(x_i + r)\sin\varphi + y_i\cos\varphi - \rho]^2\, dr \\ &= (x_i \sin\varphi + y_i\cos\varphi - \rho)^2 + \frac{1}{12} r_i^2 \sin^2\varphi \\ &= e_i^2(\rho, \varphi) + \frac{1}{12} r_i^2 \sin^2\varphi \qquad (22)\end{aligned}$$

where $e_i(\rho, \varphi)$ is the distance of the centroid of E_i to the baseline. The quantity ϵ_i^2 is essentially the mean squared distance from the edge points to the baseline L.

To obtain the Bayesian estimates, we could set the partial derivatives of $Q(\rho, \varphi)$ with respect to ρ and φ to zero and then solve the equations. Unfortunately, it can be shown that the equations do not have a closed-form solution. They can only be solved numerically using some iterative schemes, such as the gradient descent method. However, in our context, the angle φ of the baseline is very small ($\ll 10°$). It justifies the approximation $\varphi \approx \sin(\varphi)$. Let $Q(\rho, \varphi)$ be approximated by,

$$Q(\rho, \varphi) = \sum_{i=1}^{N} \frac{\epsilon_i^2}{2\sigma_\epsilon^2} + \frac{\sin^2\varphi}{2\sigma_\varphi^2}. \qquad (23)$$

The approximate Bayesian estimates of the baseline parameters $\hat{\rho}$ and $\hat{\varphi}$ can be obtained by:

$$\hat{\rho} = \mu_x \sin\hat{\varphi} + \mu_y \cos\hat{\varphi} \qquad (24)$$

$$\hat{\varphi} = -\frac{1}{2}\arctan\left(\frac{2\mu_{xy}}{\mu_{xx} - \mu_{yy} + \Sigma_{i=1}^{N} r_i^2/12N + \sigma_\epsilon^2/\sigma_\varphi^2 N}\right) \qquad (25)$$

where μ_x, μ_y, μ_{xx}, μ_{yy}, and μ_{xy} are the first- and second-order moments of the centroids of the edges.

D. Displacement Parameter Estimation. In this section, we derive the Bayesian estimate of the displacement $\hat{\mu}_\delta$ given a set of edges $E_1, E_2, \ldots, E_N$ coming from a linear displacement structure L, where L is parameterized by its baseline (ρ, φ) and the displacement μ_δ. Without the loss of generality, we assume that the edges $E_1, E_2, \ldots, E_N$ are properly ordered along the baseline direction.

Let $\delta_j = \delta(E_j, E_{j+1})$ denote the gap length between the two adjacent edges along the baseline direction, where $j = 1, 2, \ldots, N-1$.

We calculate the $\hat{\mu}_\delta$ that maximizes the conditional probability distribution

$$P(E_1, E_2, \ldots, E_N, \mu_\delta \mid \rho, \varphi) = P(\mu_\delta) \sum_{j=1}^{N-1} P(\delta_j \mid \rho, \varphi, \mu_\delta), \qquad (26)$$

which is equivalent to minimizing the following sum of squares,

$$R(\mu_\delta) = \sum_{j=1}^{N-1} \frac{(\delta_j - \mu_\delta)^2}{2\sigma_\delta^2} + \frac{(\mu_\delta - \mu)^2}{2\sigma^2}. \qquad (27)$$

Solving the minimization problem, we can obtain the optimal estimate of the displacement $\hat{\mu}_\delta$:

$$\hat{\mu}_\delta = \frac{\sigma^2 \Sigma_{j=1}^{N-1} \delta_j + \sigma_\delta^2 \mu}{(N-1)\sigma^2 + \sigma_\delta^2}. \qquad (28)$$

E. Initial Baseline Estimation. Algorithms 2 and 3 are contingent upon the availability of an initial set of baselines. In this section, we discuss a technique to obtain K potential initial baselines. It first finds K clusters of edges that correspond to the initial baselines. Then, it estimates the baseline parameters for each of the clusters.

Let $\mathscr{E} = \{E_1, E_2, \ldots, E_M\}$ denote a set of M input edges. Figure 4 shows two horizontal edges E_i and E_j. Their spatial relation can be characterized by two distances $d_x = d_x(E_i, E_j)$ and $d_y = d_y(E_i, E_j)$, which are defined as the parallel and perpendicular distances of the two edges, respectively. If the two edges overlap horizontally, then the parallel distance d_x is designated as zero.

To cluster the edges, we need the probability distribution of d_x and d_y given that two edges are adjacent in a linear displacement structure. Using a model that the parallel and perpendicular distances are independent, we have

$$P(d_x, d_y \mid \text{adjacent}) = P(d_x \mid \text{adjacent}) P(d_y \mid \text{adjacent}). \qquad (29)$$

We further assume that $P(d_x \mid \text{adjacent})$ be the Gaussian distribution and $P(d_y \mid \text{adjacent})$ be the exponential distribution, i.e.,

$$\begin{aligned} P(d_x \mid \text{adjacent}) &= \frac{1}{\sqrt{2\pi}\sigma_x} e^{-\frac{(d_x - \mu_x)^2}{2\sigma_x^2}}, \\ P(d_y \mid \text{adjacent}) &= \frac{1}{\sigma_y} e^{-\frac{d_y}{\sigma_y}} \end{aligned} \qquad (30)$$

where μ_x and σ_x^2 are the mean and variance of the parallel distance and σ_y is the perpendicular distance decay factor.

We also need the probability distribution of d_x and d_y given that two edges are nonadjacent. In this case, we assume d_x and d_y have a uniform distribution, i.e.,

$$P(d_x, d_y \mid \text{nonadjacent}) = 1/\tau. \qquad (31)$$

where τ is a constant chosen by the user.

Based on the two probability distributions, we could define an undirected graph $\mathscr{G}$. The vertices of $\mathscr{G}$ consist of all the elements in $\mathscr{E}$. The edges of $\mathscr{G}$ come from a subset of $\mathscr{E} \times \mathscr{E}$. There is an edge between two vertices E_i and E_j if and only if $P(d_x, d_y \mid \text{adjacent}) > 1/\tau$. Typically, two edges E_i and E_j will satisfy the relation if both d_x and d_y are small relative to the choice of τ.

Suppose the graph $\mathscr{G}$ has K connected subgraphs, denoted by

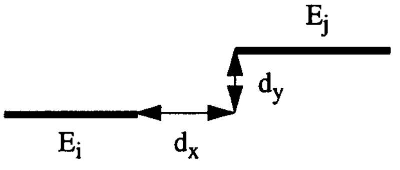


Figure 4. Parallel and perpendicular distances between the two edges. d_x is the parallel distance; d_y is the perpendicular distance.

$\mathcal{G}_k^0$, where $k = 1, 2, \ldots, K$. They can be computed via a simple depth-first search algorithm. Each of the subgraph $\mathcal{G}_k^0$ defines a cluster of edges (denoted by $\mathcal{S}_k^0$, $k = 1, 2, \ldots, K$) that correspond to an initial baseline. Without the loss of generality, we assume that $\mathcal{S}_1^0, \mathcal{S}_2^0, \ldots, \mathcal{S}_K^0$ are sorted in descending order according to their numbers of edges.

Using the method described in Section 4.3, we could then estimate the baseline parameters of the kth cluster $\mathcal{S}_k^0$. Let it be denoted as (ρ_k^0, φ_k^0), where $k = 1, 2, \ldots, K$.

F. Multiple Linear Displacement Structure Detection. In a given set of edges, there may exist *multiple linear displacement* structures. For example, it is likely that there are multiple text lines on a document image. Each text line can be modeled by a linear displacement structure. Hence, we need to construct an algorithm to find all the linear displacement structures in $\mathcal{E}$. In this section, we describe a greedy algorithm (Algorithm 4) to find multiple linear displacement structures in a set of edges. The idea is to iteratively detect and then remove one linear displacement structure from a set of edges until no more structures can be found.

Algorithm 4. Multiple Linear Displacement Structure Detection

```
1. Compute the initial baseline estimates from $\mathcal{E}$ (see Section IVE). Let the detected initial baselines be denoted by $(\rho_k^0, \varphi_k^0)$, where $k = 1, 2, \ldots, K$.
2. Let $\mathcal{E}_0 = \mathcal{E}$ and $N = 0$.
3. For $k = 1, 2, \ldots, K$, compute:
   (a) Let $\mathcal{E}_k = \mathcal{E}_{k-1}$.
   (b) Given the initial baseline $(\rho_k^0, \varphi_k^0)$ and the set of edges $\mathcal{E}_k$, find the set of collinear edges $\mathcal{S}'_k$ through Algorithm 2.
   (c) If $S'_k + \emptyset$, then repeat Step 3.
   (d) Given the set of collinear edges $\mathcal{S}'_k$, find the optimal subset of equally spaced edges $\hat{\mathcal{S}}_k$ through Algorithm 3.
       • If $\hat{\mathcal{S}}_k \neq \emptyset$, then it constitutes a linear displacement structure, denoted by $\mathcal{L}_N = \hat{\mathcal{S}}_k$. Let $N = N + 1$, $\mathcal{E}_k = \mathcal{E}_k - \hat{\mathcal{S}}_k$, and $\mathcal{S}'_k = \mathcal{S}'_k - \hat{\mathcal{S}}_k$. Repeat Step 3d.
       • Else repeat Step 3.
4. If $\mathcal{E}_k \neq \emptyset$, then the edges in $\mathcal{E}_k$ do not belong to any of the well-defined linear displacement structures. Each of them is regarded as a linear displacement structure with a single edge, i.e., for all $E \in \mathcal{E}_k$, let $N = N + 1$ and $\mathcal{L}_N = \{E\}$.
5. Calculate $\mathcal{L} = \bigcup_{n=0}^{N-1} \mathcal{L}_n$, which is the set of detected linear displacement structures from $\mathcal{E}$. □
```

V. TEXT LINE DETECTION

Theoretically, we could directly apply Algorithm 4 to find the text lines. But we discovered in some cases that the algorithm incorrectly merged two text lines from two neighboring text columns. The reason is that the two text lines share a common baseline and the algorithm is not able to discriminate two types of word gaps along the baseline, i.e., the word gap within a text line (denoted as Class I) and the word gap that jumped across two neighboring text columns (denoted as Class II).

To solve the problem, we observe that there are additional information that can be used for the discrimination. The Class II word gaps are usually accompanied with long vertical edges that correspond to either the left or the right sides of the text columns, whereas the Class I word gaps typically have very short vertical edges. Figure 5 illustrates two sets of word-bounding boxes $\mathcal{B} = \{B_1, B_2, B_3, B_4, B_5\}$ and $\mathcal{C} = \{C_1, C_2, C_3, C_4\}$ from two adjacent text columns. The right vertical edges of $\mathcal{B}$ can be modeled by a linear displacement structure L_r. Let its height be denoted as d_r, which is the height of the minimal bounding box that encloses $\mathcal{B}$. Similarly, the left vertical edges of $\mathcal{C}$ is modeled by a linear displacement structure L_l. Let its height be denoted as d_l. Then each of the words in $\mathcal{B}$ is given a right vertical edge length of d_r; and each of the words in $\mathcal{C}$ is given a left vertical edge length of d_l. For those words whose right vertical edges do not form a linear displacement structure, their right vertical edge lengths are denoted as $d_r = 0$. In the same way, for those words whose left vertical edges do not form a linear displacement structure, their left vertical edge lengths are denoted as $d_l = 0$.

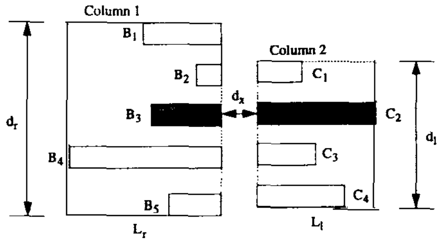


Figure 5. Left and right vertical edge lengths d_l and d_r of a word.

In the Figure 5, if we assume that B_3 and C_2 share a common baseline, then the word gaps between the two can be characterized by the horizontal displacement $\delta = d_x$ and the vertical edge length $\nu = \max(d_r, d_l)$.

In general, following the same notations as in Section IVA, let $\nu_j = \max[d_r(B_{i_j}), d_l(B_{i_{(j+1)}})]$, where $d_r(B_{i_j})$ and $d_l(B_{i_{(j+1)}})$ denote the right and left vertical edge lengths of the i_jth and $i_{(j+1)}$th words, respectively. Assume that ν_j for $j = 1, 2, \ldots, N-1$ are independently drawn from the following exponential distribution,

$$P(\nu \mid \rho, \varphi) = \frac{1}{\sigma_\nu} e^{-\frac{\nu}{\sigma_\nu}} \tag{32}$$

where σ_ν is a constant, which has the meaning of the mean vertical edge length.

By using the model that the δ and ν are independent, we can modify Equation (10) as

$$P(\mathcal{S} \mid \text{from } L) = \prod_{i \in \mathcal{I}} P(\epsilon_i \mid \rho, \varphi) \prod_{j=1}^{N-1} P(\delta_j \mid \rho, \varphi, \mu_\delta) \times \prod_{j=1}^{N-1} P(\nu_j \mid \rho, \varphi)\,. \tag{33}$$

Accordingly, the optimization terms in Equations (13) and (18) could be rewritten as

$$J(\mathcal{I}, \rho, \varphi, \mu_\delta \mid \mathcal{E}) = \sum_{i \in \mathcal{I}} \frac{\epsilon_i^2}{2\sigma_\epsilon^2} + \sum_{i \in \mathcal{I}} \frac{(\delta_i - \mu_\delta)^2}{2\sigma_\delta^2} + \sum_{i \in \mathcal{I}} \frac{\nu_i}{\sigma_\nu} + \frac{\varphi^2}{2\sigma_\varphi^2} + \frac{(\mu_\delta - \mu)^2}{2\sigma^2} - N \ln q - (M - N) \ln \gamma(1 - q) \tag{34}$$

and

$$J_2(\mathcal{I}, \mu_\delta \mid \mathcal{E}, \mathcal{I}', \hat{\rho}, \hat{\varphi}) = \sum_{i \in \mathcal{I}} \frac{(\delta_i - \mu_\delta)^2}{2\sigma_\delta^2} + \sum_{i \in \mathcal{I}} \frac{\nu_i}{\sigma_\nu} + \frac{(\mu_\delta - \mu)^2}{2\sigma^2} - N \ln q - (N' - N) \ln \gamma(1 - q) . \quad (35)$$

Consequently, Equation (19) in Algorithm 3 needs to be revised as:

$$P(u, N) = \prod_{j=1}^{N} P(\delta_j \mid \hat{\rho}, \hat{\varphi}, \hat{\mu}_\delta) \prod_{j=1}^{N} P(\nu_j \mid \rho, \varphi) q^N \cdot [\gamma(1 - q)]^{N' - N} \cdot P(\hat{\mu}_\delta) . \quad (36)$$

By approximating $\sigma_\nu \to \infty$, the solutions to optimizing Equations 36 and 19 converge because the effect of the vertical edge length ν diminishes. Therefore, the modified algorithm is a generalization of the original one.

To summarize, our text line detection algorithm is constructed in three passes. In the first two passes, the algorithm computes linear displacement structures along the vertical directions based on Algorithm 4 (let $\sigma_\nu = \infty$). The first pass detects linear displacement structures from the left vertical edges of the word bounding boxes. Each word is given a left vertical edge length d_l. Similarly, the second pass finds linear displacement structures from the right vertical edges of the word bounding boxes. Each word is given a right vertical edge length d_r. In the final pass, the algorithm runs Algorithm 4 on the set of horizontal word edges (either top, center, or bottom edges) by taking into account the vertical edge length information obtained from the previous two passes.

Algorithm 5 describes the text line detection algorithm. The input are $\hat{\Sigma}$ and an initial set of text blocks Φ^t, where denote $\hat{\Sigma} = \{B_1, B_2, \ldots, B_M\}$ and $\Phi^t = \{P_1, P_2, \ldots, P_K\}$. The output is a set of text line bounding boxes Λ^{t+1}.

Algorithm 5. Text Line Detection

1. Compute inclusion relationship between text blocks and words: Let $\{\hat{\Sigma}_k \mid k = 1, 2, \ldots, K\}$ be a partition of $\hat{\Sigma}$ and

   $$\hat{\Sigma}_k = \{B \in \hat{\Sigma} \mid k = arg \max_{i=1}^{K} Area(B \cap P_i)\}$$

   where the function $Area(B \cap P_i)$ returns the overlapped area of B by P_i.
2. For each subset of words $\hat{\Sigma}_k$ that corresponds to a text block, compute text lines in the text block by:
   (a) Pass 1: Run Algorithm 4 (choose $\sigma_\nu = \infty$) to detect linear displacement structures on the left edges of words in $\hat{\Sigma}_k$ and assign a left vertical edge length to each word.
   (b) Pass 2: Run Algorithm 4 (choose $\sigma_\nu = \infty$) to detect linear displacement structures on the right edges of words in $\hat{\Sigma}_k$ and assign a right vertical edge length to each word.
   (c) Pass 3: Run Algorithm 4 to detect linear displacement structures on the horizontal edges (top, center, or bottom edges) of words in $\hat{\Sigma}_k$. Each of the structures constitutes a detected text line. Output the bounding boxes of the detected text lines.
3. End □

Figure 6 illustrates the text line detection process. Figure 6a shows a set of extracted word-bounding boxes. Initially, we choose Φ^0 to include all the word-bounding boxes. The algorithm works itself in a sort of bootstrapping mode to obtain the initial estimates of text lines from the input word bounding boxes. Figure 6b plots the detected text lines using Algorithm 5, where the algorithm parameters are defined as follows: $\sigma_\epsilon = 8.0$, $\sigma_\varphi = 0.3$, $\mu = 21.0$, $\sigma = 10.0$, $\sigma_\delta = 10.0$, $\sigma_\nu = 50.0$, $q = 0.5$, and $\gamma = 0.1$.

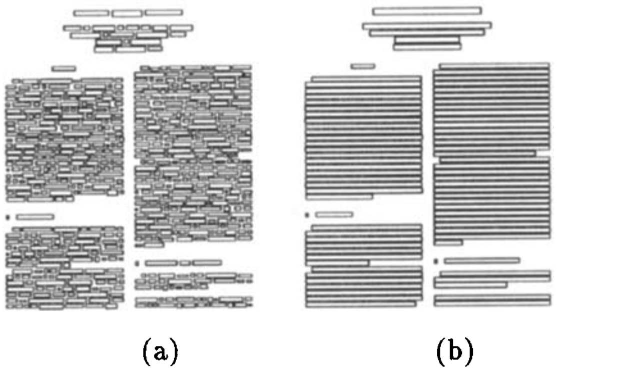


Figure 6. Text line detection process from the word-bounding boxes without the text block delineation. (a) Input word-bounding boxes; (b) output line-bounding boxes.

From Figure 6b, we notice that some section headings are detected as two separate parts instead of one. This is because the gaps between the separated parts are too large to be considered the word gaps. The problem can be overcome via the iterations in Algorithm 1. That is, we proceed to detect text blocks based on the imperfect text lines. Then, by knowing the text blocks, we will recompute the text lines.

Figure 7 illustrates the text line detection process given a set of precomputed text blocks. The uses of the algorithm under this mode could be as follows. 1) If we know ahead of time where the texts should appear on document images (usually called the text fields), then the algorithm can be used to locate text lines in the text fields. 2) If we could precompute the locations of text blocks or columns via other methods, the algorithm can find text lines in the text blocks or text columns robustly. 3) We can embed the algorithm as an intermediate step in Algorithm 1.

Figure 7a shows a set of text block bounding boxes overlaid on top of the word-bounding boxes. Figure 7b plots the detected text lines using Algorithm 5, where the algorithm parameters are defined as follows: $\sigma_\epsilon = 8.0$, $\sigma_\varphi = 0.3$, $\mu = 21.0$, $\sigma = 20.0$, $\sigma_\delta = 20.0$, $\sigma_\nu = \infty$, $q = 0.5$, and $\gamma = 0.1$. We see that all the text lines are correctly detected.

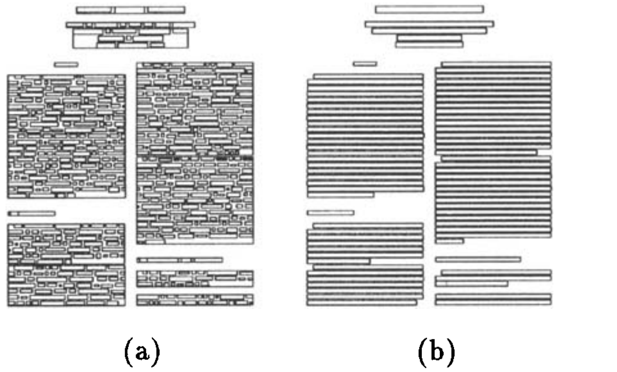


Figure 7. Text line detection process from the word-bounding boxes with the text block delineation. (a) Input word-bounding boxes; (b) output line-bounding boxes.

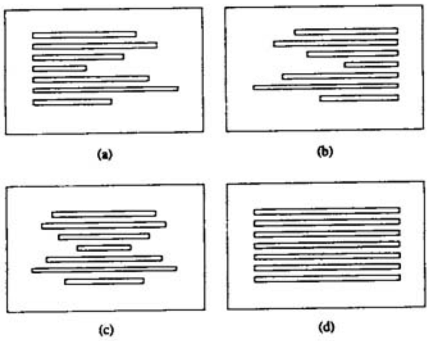


Figure 8. Four homogeneous types of alignment. (a) Left justified, (b) right justified, (c) center justified, and (d) justified. The vertical edges of their text lines can be directly modeled by the PLDM.

VI. AUGMENTED PROBABILISTIC LINEAR DISPLACEMENT MODEL

Figure 8 illustrates four homogeneous types of text block alignment. The left, center, or right vertical edges of the text line bounding boxes can be simply modeled by the PLDM.

When people refer to the *justified* texts, they also mean its variations, as shown in Figure 9. The differences are the various types of indentation at the first and/or last text lines. We could use an alternate name called *justified-indent* to designate these types of alignment.

Other types of alignment, such as *justified-hanging* and *left-hanging*, can also appear in some document images. For the most part, they can be modeled by the PLDM except the first and last line structures (Fig. 10), which is similar to the justified-indent-type alignment.

To describe these structural variations in text blocks, we introduce an augmented probabilistic linear displacement model (APLDM). Our viewpoint is based on the idea of partitioning the Λ into maximal subsets the satisfy certain statistical model constraints.

Let $\langle L_1, L_2, \ldots, L_n \rangle$ be a sequence of n consecutive text lines that come from a text block $P \in \Phi$. Let $L_i \in \Lambda$, where $i = 0, 1, \ldots, N$. Let $L_{n+1} \in \Lambda$ be another observed text line. As a convention, $\langle L_1, L_2, \ldots, L_n \rangle$ could be either descending or ascending according to their row positions in the iamge. In the descending case, L_{i+1} is below L_i, whereas in the ascending case, L_{i+1} is above L_i.

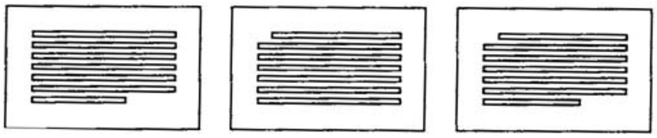


Figure 9. Variations of the justified texts. (a) Indentation at the last line; (b) indentation at the first line; (c) indentation at the first and the last lines.

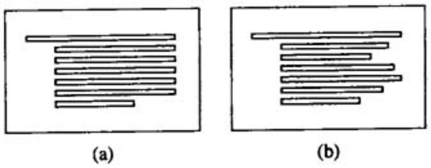


Figure 10. Hanging-type alignments. (a) Justified-hanging; (b) left-hanging.

To setup the framework, we consider the null hypothesis (denoted by $\mathcal{H}_0$) that L_{n+1} is the text line in P that immediately follows the sequence $\langle L_1, L_2, \ldots, L_n \rangle$. Under the null hypothesis $\mathcal{H}_0$, there is a probability distribution of L_{n+1} by observing the sequence $\langle L_1, L_2, \ldots, L_n \rangle$. Let it be denoted as $P(L_{n+1} \mid \langle L_1, L_2, \ldots, L_n \rangle)$.

To characterize the $P(L_{n+1} \mid \langle L_1, L_2, \ldots, L_n \rangle)$, we define the features that describe the relationships among the set of text lines. In Figure 11, let h_i and w_i denote the height and width of the text line L_i, respectively, where $i = 1, 2, \ldots, n + 1$.

Let s_i denote the line spacing between two consecutive text lines L_i and L_{i+1}, which is the gap distance between the center horizontal edges of the two lines, where $i = 1, 2, \ldots, n$.

Let l_i, c_i, and r_i denote correspondingly the distances of the left, center and right vertical edges of L_i to the left, center, and right vertical edges of the text block P (denoted by $\mathscr{E}_l$, $\mathscr{E}_c$, and $\mathscr{E}_r$, respectively), where $i = 1, 2, \ldots, n + 1$. The $\mathscr{E}_l$, $\mathscr{E}_c$, and $\mathscr{E}_r$ are estimated by fitting Bayesian straight lines on the left, center, and right vertical edges of the first n lines $L_1, L_2, \ldots, L_n$ (see Section IVC). Let $\mathscr{E}_l = (\rho_l, \varphi_l)$, $\mathscr{E}_c = (\rho_l, \varphi_c)$, and $\mathscr{E}_r = (\rho_r, \varphi_r)$.

To simplify the notation, denote $H = (h_1, h_2, \ldots, h_n)$, $W = (w_1, w_2, \ldots, w_n)$, $S = (s_1, s_2, \ldots, s_{n-1})$, $L = (l_1, l_2, \ldots, l_n)$, $C = (c_1, c_2, \ldots, c_n)$, and $R = (r_1, r_2, \ldots, r_n)$. Let $\vec{v}_i = (l_i, c_i, r_i, w_i)'$ be a column vector, where $i = 1, 2, \ldots, n$. Using the model that the text line height and the text line spacing are independent with the rest of the variables, we can write the probability $P(L_{n+1} \mid \langle L_1, L_2, \ldots, L_n \rangle)$ as

$$\begin{aligned} P(L_{n+1} \mid \langle L_1, L_2, \ldots, L_n \rangle) &= P(h_{n+1}, s_n, \vec{v}_{n+1} \mid H, S, L, C, R, W) \\ &= P(h_{n+1} \mid H) P(s_n \mid S) P(\vec{v}_{n+1} \mid L, C, R, W) \,. \quad (37) \end{aligned}$$

Furthermore, we assume that the text line height and the text line spacing are i.i.d. (i.e., identically and independently distributed) Gaussian with unknown means and known variances. Let μ_h and σ_h^2 denote the mean and variance of the text line height. Let μ_s and σ_s^2 denote the mean and variance of the text line spacing. Then, by first-order approximation, the probabilities $P(h_{n+1} \mid H)$ and $P(s_n \mid S)$ can be approximated by,

$$\begin{aligned} P(h_{n+1} \mid H) &= \int_{\mu_h} P(h_{n+1}, \mu_h \mid H) \, d\mu_h \\ &= \int_{\mu_h} P(h_{n+1} \mid \mu_h, H) P(\mu_h \mid H) \, d\mu_h \\ &= \int_{\mu_h} P(h_{n+1} \mid \mu_h) P(\mu_h \mid H) \, d\mu_h \end{aligned}$$

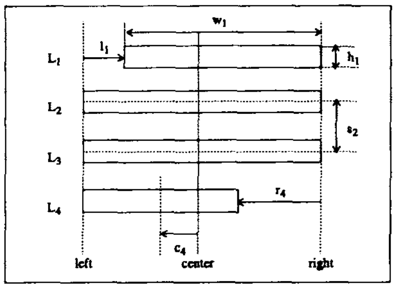


Figure 11. Text line features.

$$\approx P(h_{n+1} \mid \hat{\mu}_h) = \frac{1}{\sqrt{2\pi}\sigma_h} e^{-\frac{(h_{n+1}-\hat{\mu}_h)^2}{2\sigma_h^2}} \tag{38}$$

and

$$\begin{aligned} P(s_n \mid S) &= \int_{\mu_s} P(s_n, \mu_s \mid S)\, d\mu_s \\ &= \int_{\mu_s} P(s_n \mid \mu_s, S) P(\mu_s \mid S)\, d\mu_s \\ &= \int_{\mu_s} P(s_n \mid \mu_s) P(\mu_s \mid S)\, d\mu_s \\ &\approx P(s_n \mid \hat{\mu}_s) = \frac{1}{\sqrt{2\pi}\sigma_s} e^{-\frac{(s_n-\hat{\mu}_s)^2}{2\sigma_s^2}} \end{aligned} \tag{39}$$

where $\hat{\mu}_h$ is the estimaed mean text line height that maximizes $P(\mu_h \mid H)$, and $\hat{\mu}_s$ is the estimated mean text line spacing that maximizes $P(\mu_s \mid S)$. We assume that μ_h has a Gaussian $N(\bar{\mu}_h, \sigma^2_{\bar{\mu}_h})$ prior probability distribution and that μ_s has a Gaussian $N(\bar{\mu}_s, \sigma^2_{\bar{\mu}_s})$ prior probability distribution. Then, the $\hat{\mu}_h$ and $\hat{\mu}_s$ can be computed through Equations (40) and (41) (see Section IVD):

$$\hat{\mu}_h = \frac{\sigma^2_{\bar{\mu}_h} \sum_{i=1}^{n} h_i + \sigma_h^2 \bar{\mu}_h}{n\sigma^2_{\bar{\mu}_h} + \sigma_h^2} \tag{40}$$

$$\hat{\mu}_s = \frac{\sigma^2_{\bar{\mu}_s} \sum_{i=1}^{n-1} s_i + \sigma_s^2 \bar{\mu}_s}{(n-1)\sigma^2_{\bar{\mu}_s} + \sigma_s^2}. \tag{41}$$

If $\langle L_1, L_2, \ldots, L_n \rangle$ is descending, then L_{n+1} can be either the intermediate or the last line of P. Let X be a random binary variable. It has a binary one value if L_{n+1} is an intermediate line and has a binary zero value if L_{n+1} is the last line. Let Z denote the type of alignment of the text block P. It can take values such as "justified" ($Z = 0$), "left-justified" ($Z = 1$), "right-justified" ($Z = 2$), and "center-justified" ($Z = 3$). Then, the joint probability distribution $P(\vec{v}_{n+1} \mid L, C, R, W)$ can be evaluated as:

$$\begin{aligned} &P(\vec{v}_{n+1} \mid L, C, R, W) \\ &= \sum_{x=0}^{1} \sum_{z=0}^{3} P(\vec{v}_{n+1}, x, z \mid L, C, R, W) \\ &= \sum_{x=0}^{1} \sum_{z=0}^{3} P(\vec{v}_{n+1} \mid x, z, L, C, R, W) P(x \mid z, L, C, R, W) \\ &\quad \times P(z \mid L, C, R, W) \\ &= \sum_{x=0}^{1} \sum_{z=0}^{3} P(\vec{v}_{n+1} \mid x, z, L, C, R, W) P(x) P(z \mid L, C, R, W) \end{aligned} \tag{42}$$

where $P(x) = P(X = x)$ is the prior probability of the line L_{n+1} being an intermediate line ($X = 1$) or the last line ($X = 0$) under the null hypothesis $\mathscr{H}_0$.

If $\langle L_1, L_2, \ldots, L_n \rangle$ is ascending, then L_{n+1} can be either the intermediate or the first line of P. Let Y be a random binary variable. It has a binary one value if L_{n+1} is an intermediate line and has a binary zero value if L_{n+1} is the last line. We use the same Z to denote the type of alignment of the text block P. Then, the joint probability distribution $P(\vec{v}_{n+1} \mid L, C, R, W)$ can be similarly evaluated as:

$$\begin{aligned} P(\vec{v}_{n+1} \mid L, C, R, W) = &\sum_{y=0}^{1} \sum_{z=0}^{3} P(\vec{v}_{n+1} \mid y, z, L, C, R, W) \\ &\times P(y) P(z \mid L, C, R, W) \end{aligned} \tag{43}$$

where $P(y) = P(Y = y)$ is the prior probability of the line L_{n+1} being an intermediate line ($Y = 1$) or the first line ($Y = 0$) under the null hypothesis $\mathscr{H}_0$.

By the Bayesian rule, $P(z \mid L, C, R, W)$ can be calculated as:

$$\begin{aligned} P(z \mid L, C, R, W) &= \frac{P(L, C, R, W \mid z) P(z)}{P(L, C, R, W)} \\ &= \frac{P(L, C, R, W \mid z) P(z)}{\sum_{z=0}^{3} P(L, C, R, W \mid z) P(z)} \end{aligned} \tag{44}$$

where $P(z)$ is the prior probability of observing the type $Z = z$ text block alignment. The probability $P(L, C, R, W \mid z)$ can be approximated by [12]:

$$\begin{aligned} P(L, C, R, W \mid z) = &\int P(L, C, R, W \mid z, \mathscr{E}_l, \mathscr{E}_c, \mathscr{E}_r, \mu_w) \\ &\times P(\mathscr{E}_l, \mathscr{E}_c, \mathscr{E}_r, \mu_w \mid z)\, d\rho_l\, d\rho_c\, d\rho_r\, d\varphi_l\, d\varphi_r\, d\mu_w \\ &\approx P(L, C, R, W \mid z, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w) \\ &= \prod_{i=1}^{n} P(\vec{v}_i \mid z, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w) \end{aligned} \tag{45}$$

where $\hat{\mathscr{E}}_l$, $\hat{\mathscr{E}}_c$, and $\hat{\mathscr{E}}_r$ are the Bayesian estimates of the $\mathscr{E}_l$, $\mathscr{E}_c$, and $\mathscr{E}_r$, respectively. The $\hat{\mu}_w = \frac{1}{n} \sum_{i=1}^{n} w_i$ is the sample mean text line width. The random variables ρ_l, ρ_c, ρ_r, and μ_w have uniform prior probability distributions.

Since $\langle L_1, L_2, \ldots, L_n \rangle$ are assumed to be intermediate text lines, they come from one of the four homogeneous types of alignment as illustrated in Figure 8. The probabilities $P(\vec{v}_i \mid z, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w)$ for $z = 0, 1, 2, 3$ have the following form:

$$\begin{aligned} P(\vec{v}_i \mid Z = 0, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w) &= P(c_i, w_i \mid Z = 0, \hat{\mathscr{E}}_c, \hat{\mu}_w) \\ &= P(c_i \mid Z = 0, \hat{\mathscr{E}}_c) P(w_i \mid Z = 0, \hat{\mu}_w) \\ &= \frac{1}{\sqrt{2\pi}\sigma_\epsilon} e^{-\frac{c_i^2}{2\sigma_\epsilon^2}} \cdot \frac{1}{\sqrt{2\pi}\sigma_w} e^{-\frac{(w_i-\hat{\mu}_w)^2}{2\sigma_w^2}} \end{aligned} \tag{46}$$

$$\begin{aligned} P(\vec{v}_i \mid Z = 1, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w) &= P(l_i, w_i \mid Z = 1, \hat{\mathscr{E}}_l, \hat{\mu}_w) \\ &= P(l_i \mid Z = 1, \hat{\mathscr{E}}_l) P(w_i \mid Z = 1, \hat{\mu}_w) \\ &= \frac{\lambda}{\sqrt{2\pi}\sigma_\epsilon} e^{-\frac{l_i^2}{2\sigma_\epsilon^2}} \end{aligned} \tag{47}$$

$$\begin{aligned} P(\vec{v}_i \mid Z = 2, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w) &= P(r_i, w_i \mid Z = 2, \hat{\mathscr{E}}_r, \hat{\mu}_w) \\ &= P(r_i \mid Z = 2, \hat{\mathscr{E}}_r) P(w_i \mid Z = 2, \hat{\mu}_w) \\ &= \frac{\lambda}{\sqrt{2\pi}\sigma_\epsilon} e^{-\frac{r_i^2}{2\sigma_\epsilon^2}} \end{aligned} \tag{48}$$

$$\begin{aligned} P(\vec{v}_i \mid Z = 3, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w) &= P(c_i, w_i \mid Z = 3, \hat{\mathscr{E}}_c, \hat{\mu}_w) \\ &= P(c_i \mid Z = 3, \hat{\mathscr{E}}_c) P(w_i \mid Z = 3, \hat{\mu}_w) \\ &= \frac{\lambda}{\sqrt{2\pi}\sigma_\epsilon} e^{-\frac{c_i^2}{2\sigma_\epsilon^2}} \end{aligned} \tag{49}$$

Table I. Probability distributions of $P(\vec{v}_{n+1} \mid x, z, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w)$.

	Descending Text Line Sequence	
$Z = z$	$X = 0$	$X = 1$
0	$P(\vec{v}_{n+1} \mid Z = 1, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w)$	$P(\vec{v}_{n+1} \mid Z = 0, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w)$
1	$P(\vec{v}_{n+1} \mid Z = 1, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w)$	$P(\vec{v}_{n+1} \mid Z = 1, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w)$
2	$P(\vec{v}_{n+1} \mid Z = 2, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w)$	$P(\vec{v}_{n+1} \mid Z = 2, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w)$
3	$P(\vec{v}_{n+1} \mid Z = 3, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w)$	$P(\vec{v}_{n+1} \mid Z = 3, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w)$

where σ_ϵ^2 and σ_w^2 are known variances. Let λ be a constant. For the left-, right-, and center-justified text, we assume that the text line width follows a uniform distribution.

In the same way, we can approximate the probabilities $P(\vec{v}_{n+1} \mid x, z, L, C, R, W)$ and $P(\vec{v}_{n+1} \mid y, z, L, C, R, W)$ as follows:

$$P(\vec{v}_{n+1} \mid x, z, L, C, R, W) \approx P(\vec{v}_i \mid x, z, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w) \quad (50)$$

$$P(\vec{v}_{n+1} \mid y, z, L, C, R, W) \approx P(\vec{v}_i \mid y, z, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w) \quad (51)$$

Tables I and II summarize the distributions $P(\vec{v}_{n+1} \mid x, z, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w)$ and $P(\vec{v}_{n+1} \mid y, z, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w)$ for the descending and ascending text line sequences.

To summarize, we can compute the probabilities in Equations (42) and (43) as Equations (52) and (53) respectively:

$$\begin{aligned} P(\vec{v}_{n+1} \mid L, C, R, W) &= [P(\vec{v}_{n+1} \mid Z = 1, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w) P(X = 0) \\ &\quad + P(\vec{v}_{n+1} \mid Z = 0, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w) P(X = 1)] P(Z = 0 \mid L, C, R, W) \\ &\quad + \sum_{z=1}^{3} P(\vec{v}_{n+1} \mid z, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w) P(z \mid L, C, R, W) \end{aligned} \quad (52)$$

$$\begin{aligned} P(\vec{v}_{n+1} \mid L, C, R, W) &= [P(\vec{v}_{n+1} \mid Z = 2, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w) P(Y = 0) \\ &\quad + P(\vec{v}_{n+1} \mid Z = 0, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w) P(Y = 1)] P(Z = 0 \mid L, C, R, W) \\ &\quad + \sum_{z=1}^{3} P(\vec{v}_{n+1} \mid z, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w) P(z \mid L, C, R, W) \end{aligned} \quad (53)$$

Therefore, for any given L_{n+1}, we can compute the following test statistics:

$$\tau(n) = -\ln[P(L_{n+1} \mid \langle L_1, L_2, \ldots, L_n \rangle)] \,. \quad (54)$$

Let $P[\tau(n) \mid \mathscr{H}_0]$ denote the probability distribution of $\tau(n)$ under the null hypothesis $\mathscr{H}_0$. Let $P[\tau(n) \mid \mathscr{H}_1]$ denote the probability distribution of $\tau(n)$ under the alternate hypothesis $\mathscr{H}_1$. Then, we could set a threshold $T(n)$ to minimize the false-alarm rate and the misdetection rate. The decision rule would be as follows: if $\tau(n) \le T(n)$, then decide on the null hypothesis $\mathscr{H}_0$; else, decide on the alternate hypothesis $\mathscr{H}_1$. The two distributions $P[\tau(n) \mid \mathscr{H}_0]$ and $P[\tau(n) \mid \mathscr{H}_1]$ will be determined through the experiments (see Section VIIIB).

Table II. Probability distributions of $P(\vec{v}_{n+1} \mid y, z, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w)$.

	Ascending Text Line Sequence	
$Z = z$	$Y = 0$	$Y = 1$
0	$P(\vec{v}_{n+1} \mid Z = 2, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w)$	$P(\vec{v}_{n+1} \mid Z = 0, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w)$
1	$P(\vec{v}_{n+1} \mid Z = 1, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w)$	$P(\vec{v}_{n+1} \mid Z = 1, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w)$
2	$P(\vec{v}_{n+1} \mid Z = 2, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w)$	$P(\vec{v}_{n+1} \mid Z = 2, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w)$
3	$P(\vec{v}_{n+1} \mid Z = 3, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w)$	$P(\vec{v}_{n+1} \mid Z = 3, \hat{\mathscr{E}}_l, \hat{\mathscr{E}}_c, \hat{\mathscr{E}}_r, \hat{\mu}_w)$

VII. TEXT BLOCK DETECTION

Once the probability distributions $P[\tau(n) \mid \mathscr{H}_0]$ and $P[\tau(n) \mid \mathscr{H}_1]$ are known, we can easily construct an algorithm to detect the text blocks given a set of text lines. Algorithm 6 illustrates such an algorithm. Let Λ^t denote a set of input text lines. Let the output text blocks be denoted as Φ^{t+1}.

Algorithm 6. Text Block Detection

```
1. Let $X = \Lambda^t$ and $\Phi^{t+1} = \emptyset$.
2. Repeat the following steps until $X = \emptyset$:
   (a) Pick a text line from $X$. Let $x_0 \in X$ and $X = X - \{x_0\}$.
   (b) Let $Y = \{x_0\}$ and let $n$ denote the size of $Y$.
   (c) Grow $Y$ in both directions. Repeat the following steps
       until no more members of $X$ can be added to $Y$:
       • Sort the text lines in $Y$. Let $Y_d$ denote the descending
         sequence, and $Y_a$ denote the ascending sequence.
       • Calculate $P_d = P(x \mid Y_d)$ and $P_a = P(x \mid Y_a)$, which are
         the probabilities of $x$ being the bottom-most line and
         the top-most line, respectively.
       • Compute $S = \{x \in X \mid \tau(n) = -\ln[\max(P_d, P_a)] \le T(n)\}$.
       • Let $Y = Y \cup S$ and $X = X - S$.
   (d) If $n > 1$, continue to grow $Y$ downwardly. Repeat the
       following steps until not more members of $X$ can be
       added to $Y$:
       • Sort the text lines in $Y$. Let $Y_d$ denote the descending
         sequence. Remove the top-most line from $Y_d$, let it be
         denoted as $Y_d'$. The $Y_d'$ will have size $n'$.
       • Calculate $P_d = P(x \mid Y_d')$, which are the probabilities of
         $x$ being the bottom-most line.
       • Compute $S = \{x \in X \mid \tau(n) = -\ln P_d \le T(n)\}$.
       • Let $Y = Y \cup S$ and $X = X - S$.
   (e) If $n > 1$, continue to grow $Y$ upwardly. Repeat the
       following steps until no more members of $X$ can be
       added to $Y$:
       • Sort the text lines in $Y$. Let $Y_a$ denote the ascending
         sequence. Remove the bottom-most line from $Y_a$, let it
         be denoted as $Y_a'$. The $Y_a'$ will have size $n'$.
       • Calculate $P_a = P(x \mid Y_a')$, which are the probabilities of
         $x$ being the top-most line.
       • Compute $S = \{x \in X \mid \tau(n) = -\ln P_a \le T(n)\}$.
       • Let $Y = Y \cup S$ and $X = X - S$.
   (f) Calculate the bounding boxes of $Y$ and add it to $\Phi^{t+1}$.
3. Output $\Phi^{t+1}$. □
```

To illustrate the algorithm, Figure 12a shows a set of text line bounding boxes and Figure 12b plots the detected text blocks using the Algorithm 6, where we choose $T(n) = 30.0$ for all n (see Section VIIIB). We see that all the text blocks are correctly detected.

Figure 13 illustrates a degenerate case of the augmented PLDM model. Figure 13a shows the set of inputs text line bounding boxes and Figure 13b plots the detected text blocks using Algorithm 6. It shows a justified-indent text block with only two lines. In this situation, Algorithm 6 failed to compute the correct text block bounding boxes. Instead, we can implement a heuristic procedure to handle this kind of case during the postprocessing stage. Figure 13c shows the post-processing result by using the heuristics that each of the merging two text blocks contains a single text line, and that the merged text block does not intersect with other text blocks in the document image.

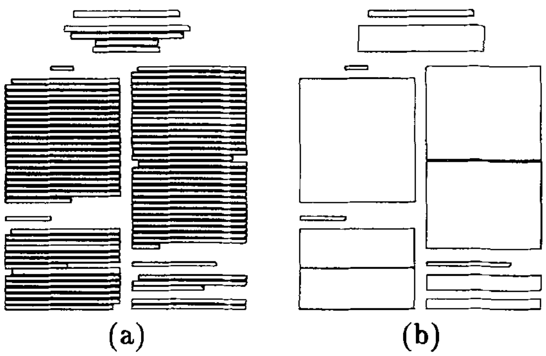


Figure 12. Text block detection process from the text line bounding boxes. (a) Input text line bounding boxes; (b) output text block bounding boxes.

VIII. EXPERIMENTAL PROTOCOL

The experiments consists of two stages. In the first stage, we conduct a series of experiments on the set of 168 fully layout ground-truthed document images described in [10]. We also rotate these images and their corresponding glyph, word, text line, and text block bounding boxes at various degrees of ±0.2°, ±0.4°, and ±0.6°. The rotation of a bounding box is done in such a way that we first rotate its four corners, and then we calculate the minimum rectilinear bounding box that encloses all the rotated corners. This generates a total population of $1176 = 168 \times 7$ ground-truthed document images. From these images, we compute the empirical probability distributions that we used to characterize the text line and text block structures. The purpose is to experimentally validate our models and show that they are in fact close approximations of the real situations. In addition, the empirical distributions provide a basis for choosing the default model parameters.

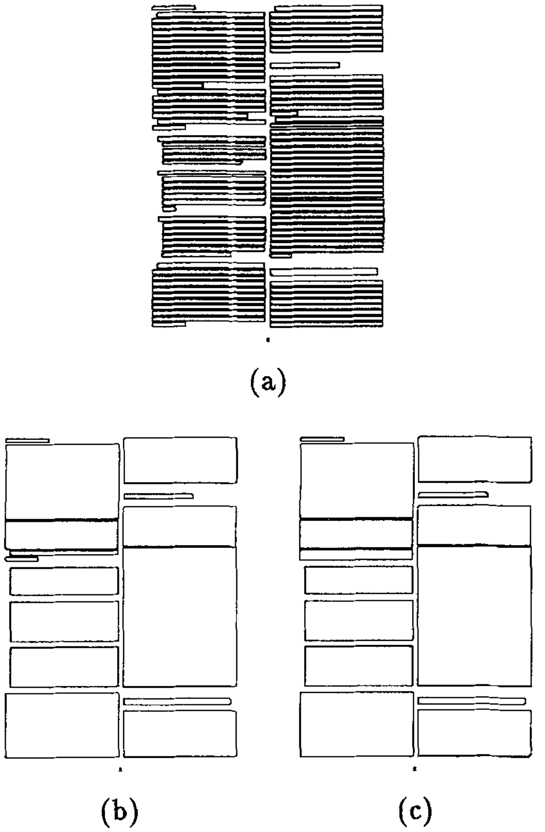


Figure 13. Degenerate case of the text block detection from the text line bounding boxes. (a) Text line bounding boxes; (b) output text block bounding boxes from Algorithm 6; (c) text block bounding boxes after postprocessing.

In the second stage, we perform experiments to evaluate the text line and text block detection algorithm quantitatively under the various configurations. The text word bounding boxes detected via the word segmentation algorithm described in [10,11] are used during the evaluation. The algorithm parameters are set to their default values obtained during the model validation processes unless otherwise indicated. We then compare the ground truth text line and text block bounding boxes with those obtained by the detection, and compute the rates of miss, false, correct, splitting, merging, and spurious detections for the text lines and text blocks [10,11].

A. Text Line Model Validation. Figure 14 plots empirical PLDM distributions for the text lines. Figure 14a shows the prior distribution of the baseline orientation φ. Figure 14b shows the distribution of the squared line-fit error ϵ^2. The center horizontal word edges has the smallest tail area than the other two. This suggests that we should use the center horizontal word edges to extract text lines. Figure 14c shows the prior distribution of the mean word displacement μ_δ within a text line. Figure 14d shows the distribution of the word displacement δ around its mean μ_δ within a text line. From the figures, we estimated $\sigma_\varphi \approx 0.6°$, $\sigma_\epsilon \approx 3.0$, $\mu \approx 21.0$, $\sigma \approx 5.0$, and $\sigma_\delta \approx 2.0$.

B. Text Block Model Validation. Figure 15 plots empirical APLDM distributions for the text blocks. Figure 15a shows the prior distribution of the mean line height μ_h within a text block. Figure 15b shows the distribution of the line height around its mean μ_h within a text block. Figure 15c shows the prior distribution of the mean line spacing μ_s within a text block. Figure 15d shows the distribution of the line spacing around its mean μ_s within a text block. Figure 15e shows the prior distribution of the mean text line width for justified text blocks (exclude the first and the last text lines). It is approximately uniform across a wide range of values. Figure 15f shows the distribution of the text line width around its mean for the justified text blocks. Figure 15g shows the distribution of the baseline orientation φ for the aligned vertical text line edges. Figure 15h shows the distribution of the squared line-fit error ϵ^2 for the aligned vertical text line edges. From the figures, we estimated $\bar{\mu}_h \approx 45.0$, $\sigma_{\bar{\mu}_h} \approx 12.0$, $\sigma_h \approx 3.0$, $\bar{\mu}_s \approx 50.0$, $\sigma_{\bar{\mu}_s} \approx 9.0$, $\sigma_s \approx 2.0$, $\sigma_w \approx 2.0$, $\sigma_\varphi \approx 0.4°$, and $\sigma_\epsilon \approx 2.0$.

Figure 16 illustrates the empirical probability distributions of $\tau(n)$ under the null hypothesis $\mathscr{H}_0$. In the experiment, we chose the following parameter values: 1) $P(Z=0) = P(Z=1) = P(Z=2) = P(Z=3)$; 2) $P(X=0) = P(X=1)$; 3) $P(Y=0) = P(Y=1)$; 4) $\sigma_\epsilon = 1.0$, $\sigma_\varphi = 0.4$, $q = 0.5$, $\gamma = 0.1$; 5) $\bar{\mu}_h = 45.0$, $\sigma_{\bar{\mu}_h} = 12.0$, $\sigma_h = 3.0$; 6) $\bar{\mu}_s = 50.0$, $\sigma_{\bar{\mu}_s} = 9.0$, $\sigma_s = 2.0$; 7) $\sigma_w = 3.0$, $\lambda = 0.05$; 8) $1 \leq n \leq 8$. We do not plot the empirical probability distributions of $\tau(n)$ under the alternate hypothesis $\mathscr{H}_1$ because it is shown that they have virtually zero probabilities in the range $0.0 \leq \tau(n) \leq 40.0$. From the figures, we observe that the variations of $P[\tau(n) \mid \mathscr{H}_0]$ with respect to n are very small, especially for $n \geq 2$. This suggests that we could use a single constant threshold T to replace the variable threshold $T(n)$ in Algorithm 6.

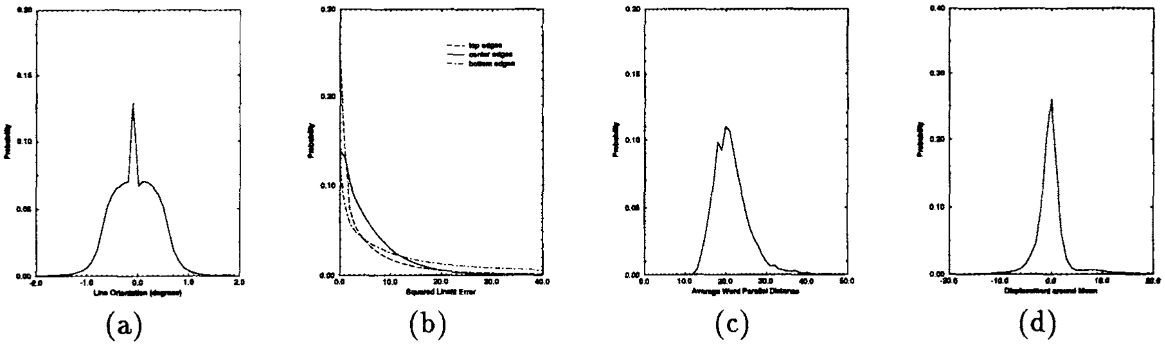


Figure 14. Empirical PLDM distributions for the text lines.

C. Performance Evaluation Results. Tables III and IV summarize the performance of the text line detection algorithm (Algorithm 5) by knowing the initial ground truth text blocks. The parameter settings are $\sigma_\epsilon = 8.0$ and $\sigma_\delta = 1000.0$. Some of the text line detection errors may be due to the imperfect word detection [10,11].

Tables V and VI summarize the performance of the text line detection algorithm (Algorithm 5) without knowing the initial text blocks. The parameter settings are $\sigma_\epsilon = 8.0$, $\sigma_\nu = 150.0$, and $\sigma_\delta = 6.0$.

Tables VII and VIII summarize the performance of the text block detection algorithm (Algorithm 6) by knowing the ground truth text lines. The parameter settings are $T(n) = 70.0$ for all n, and $\sigma_h = 9.0$.

Finally, Tables IX–XII summarize the performance of the iterative text line and text block detection algorithm (Algorithm 1), where its input are the word-bounding boxes from the word segmentation algorithm. We chose the number of iterations $N_{iter} = 7$. Tables IX and X show the rates of miss, false, correct, splitting, merging, and spurious detections for text lines with respect to the ground truth as well as the algorithm output.

On the other hand, Tables XI and XII show the rates of miss, false, correct, splitting, merging, and spurious detections for text blocks with respect to the ground truth as well as the algorithm output.

X. CONCLUSIONS AND FUTURE WORK

In this article, we discussed a statistical method for modeling and extracting text lines and text blocks from document images. We derived the so-called *probabilistic linear displacement model* (PLDM) to model the text line structures from text word bounding boxes. We also developed an augmented PLDM model to characterize the text block structures from text line bounding boxes. We gathered statistics by going through a large population of document images.

We described and evaluated an iterative text line and text block detection algorithm and reported its quantitative performance in

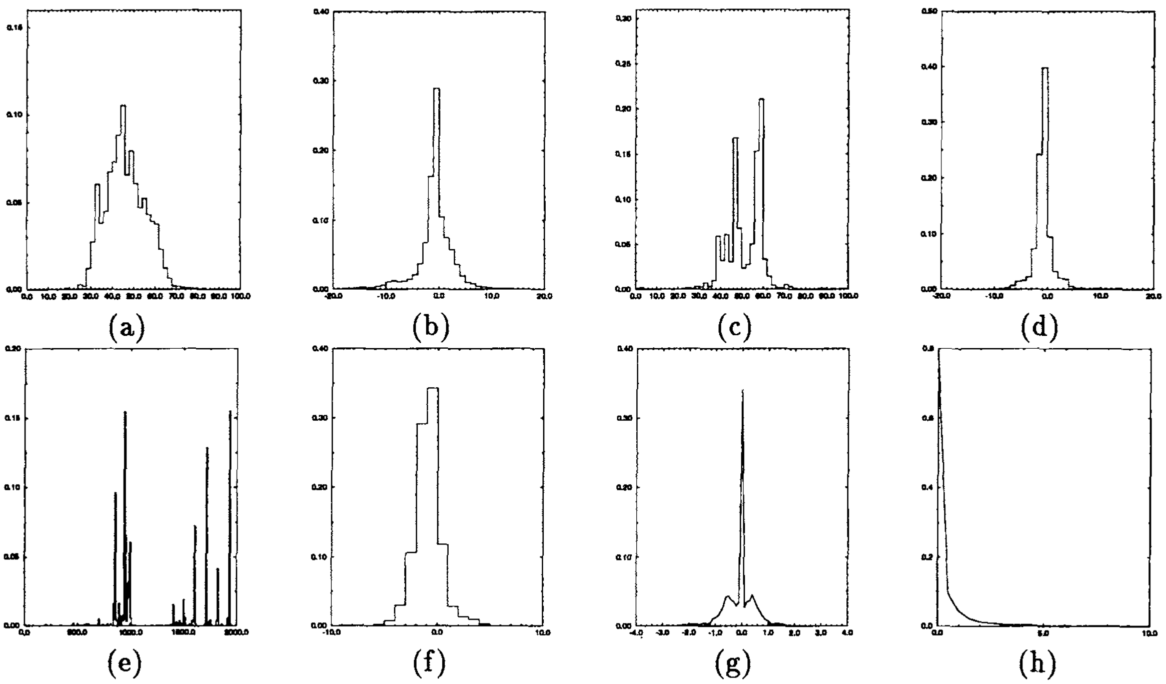
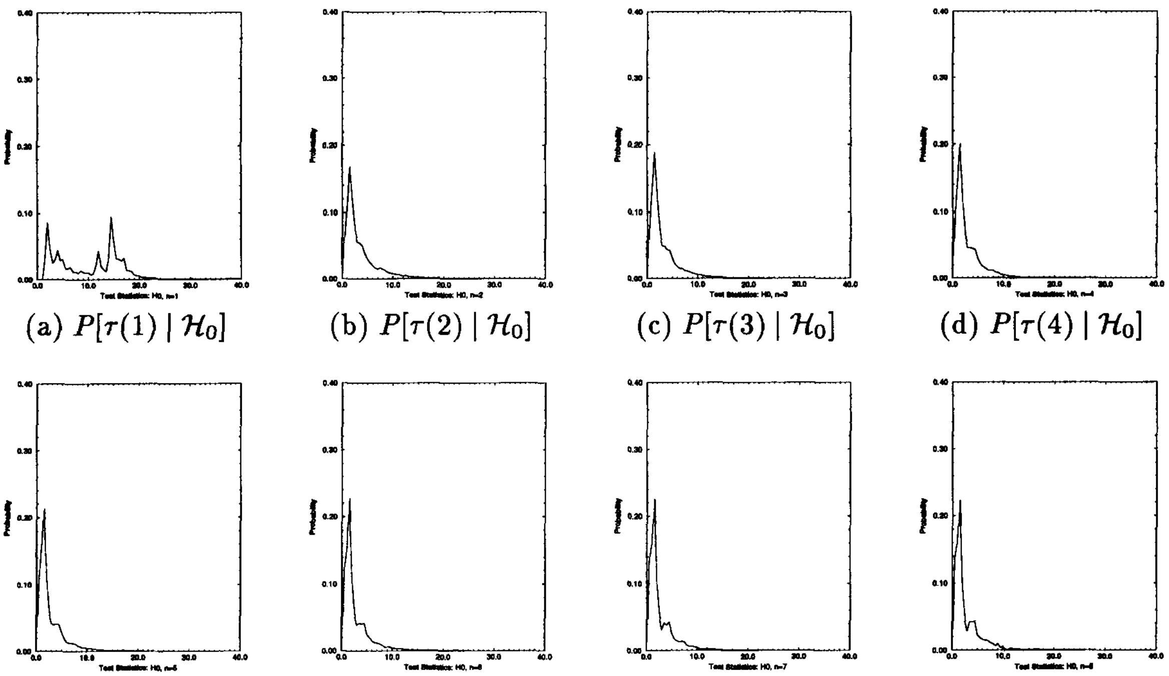


Figure 15. Empirical APLDM distributions for the text blocks.



(a) $P[\tau(1) \mid \mathcal{H}_0]$ (b) $P[\tau(2) \mid \mathcal{H}_0]$ (c) $P[\tau(3) \mid \mathcal{H}_0]$ (d) $P[\tau(4) \mid \mathcal{H}_0]$

(e) $P[\tau(5) \mid \mathcal{H}_0]$ (f) $P[\tau(6) \mid \mathcal{H}_0]$ (g) $P[\tau(7) \mid \mathcal{H}_0]$ (h) $P[\tau(8) \mid \mathcal{H}_0]$

Figure 16. Empirical distribution of the test statistics $\tau(n)$ under the null hypothesis $\mathcal{H}_0$.

Table III. Performance with respect to the ground truth text lines.

Total Ground Truth Lines	Correct	Splitting	Merging	Miss	Spurious
44,491	43,048 (96.7566%)	79 (0.1776%)	1299 (2.9197%)	7 (0.0157%)	58 (0.1304%)

Table IV. Performance with respect to the detected text lines.

Total Detected Lines	Correct	Splitting	Merging	False	Spurious
43,878	43,048 (98.1084%)	158 (0.3601%)	614 (1.3993%)	0 (0.0000%)	58 (0.1322%)

Table V. Performance with respect to the ground truth text lines.

Total Ground Truth Lines	Correct	Splitting	Merging	Miss	Spurious
44,491	40,193 (90.3396%)	3332 (7.4892%)	709 (1.5936%)	6 (0.0135%)	251 (0.5642%)

Table VI. Performance with respect to the detected text lines.

Total Detected Lines	Correct	Splitting	Merging	False	Spurious
47,888	40,193 (83.9313%)	6968 (14.5506%)	350 (0.7309%)	75 (0.1566%)	302 (0.6306%)

Table VII. Performance with respect to the ground truth text blocks.

Total Ground Truth Blocks	Correct	Splitting	Merging	Miss	Spurious
11,988	10,403 (86.7784%)	174 (1.4515%)	1044 (8.7087%)	4 (0.0334%)	363 (3.0280%)

Table VIII. Performance with respect to the detected text blocks.

Total Detected Blocks	Correct	Splitting	Merging	False	Spurious
11,657	10,403 (89.2425%)	429 (3.6802%)	458 (3.9290%)	0 (0.0000%)	367 (3.1483%)

Table IX. Performance with respect to the ground truth text lines.

Total Ground Truth Lines	Correct	Splitting	Merging	Miss	Spurious
44,491	40,193 (90.3396%)	3332 (7.4892%)	709 (1.5936%)	6 (0.0135%)	251 (0.5642%)

Table X. Performance with respect to the detected text lines.

Total Detected Lines	Correct	Splitting	Merging	False	Spurious
47,888	40,193 (83.9313%)	6968 (14.5506%)	350 (0.7309%)	75 (0.1566%)	302 (0.6306%)

Table XI. Performance with respect to the ground truth text blocks.

Total Ground Truth Blocks	Correct	Splitting	Merging	Miss	Spurious
11,988	10,403 (86.7784%)	174 (1.4515%)	1044 (8.7087%)	4 (0.0334%)	363 (3.0280%)

Table XII. Performance with respect to the detected text blocks.

Total Detected Blocks	Correct	Splitting	Merging	False	Spurious
11,657	10,403 (89.2425%)	429 (3.6802%)	458 (3.9290%)	0 (0.0000%)	367 (3.1483%)

terms of the rates of miss, false, correct, splitting, merging, and spurious detections of the text lines and text blocks.

As future work, we want to evaluate our system on the real document images from the UW English Document Image Database (I) and (II). It would first require the construction of ground truth bounding boxes for text lines and text blocks for all the images in the two databases. We also want to optimize the performance of the text line and text block detection algorithm under different parameter settings.

REFERENCES

1. F. M. Wahl, K. Y. Wong, and R. G. Casey, "Block segmentation and text extraction in mixed text image documents," *Computer Graphics and Image Processing* **20**, 375–390 (1982).
2. M. K. Krishnamoorthy, G. Nagy, S. Seth, and M. Viswanathan, "Syntactic segmentation and labeling of digitized pages from technical journals," *IEEE Trans. PAMI* **15**, (1993).
3. D. Wang and S. N. Srihari, "Classification of newspaper image blocks using texture analysis," *Computer Vision, Graphics Image Process.* **47**, 327–352 (1989).
4. L. A. Fletcher and R. Kasturi, "A robust algorithm for text string separation from mixed text/graphics images," *IEEE Trans. PAMI* **10**, 910–918 (1988).
5. A. L. Spitz, "Text characterization by connected component transformations," *Proc. SPIE* **2181**, 97–105 (1994).
6. H. S. Baird, "Background structure in document images," in *Advances in Structural and Syntactic Pattern Recognition*, World Scientific, Singapore, 1992, pp. 253–269.
7. T. Saitoh and T. Pavlidis, "Page segmentation without rectangle assumption," in *Proceedings of the 11th IAPR International Conference on Pattern Recognition*, Hague, The Netherlands, Aug. 30–Sept. 3, 1992, pp. 277–280.
8. J. Ha, I. T. Phillips, and R. M. Haralick, "Document page decomposition by the bounding-box projection techniques," in ICDAR'95: Third International Conference on Document Analysis and Recognition, Montreal, Canada, August 14–16, 1995.
9. I. T. Phillips, S. Chen, and R. M. Haralick, "English document database standard," in *Proceedings of the Second International Conference on Document Analysis and Recognition*, Japan, October 20–22, 1993, pp. 478–483.
10. S. Chen, R. M. Haralick, and I. T. Phillips, "Perfect document layout ground truth generation using DVI files and simultaneous text word detection from document images," in Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, April 1995.
11. S. Chen, R. M. Haralick, and I. T. Phillips, "Extraction of text words on document images based on a statistical characterization," *J. Electronic Imaging* **5**, 25–36 (1996).
12. S. Chen, "Document layout analysis using recursive morphological transforms," Ph.D. dissertation, University of Washington, August, 1995.
13. R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, Addison-Wesley, 1991.